1, 创建项目(掌握)

- 目的: 能够通过django的命令创建工程
- 过程:
    - 1, 进入到虚拟环境
    - 2,创建项目: django-admin startproject (项目名字)
    - 3,进入到manage.py所在的文件夹中
    - 4,启动项目: python manage.py runserver
- 注意点:
    - 如果不指定,django默认的端口8000
    - 也可以指定: python manage.py runserver ip:port

2,创建子应用(掌握)

- 目的: 能够使用djanog中的命令来创建子应用
- 过程:
    - 1, 进入到manage.py所在的同级文件中
    - 2, 创建子应用: python manage.py startapp (子应用名字)

3, 第一个helloworld程序(掌握)

- 目的: 能够在子应用中编写视图函数helloworld
- 过程:
    - 1, 在views文件中编写视图函数

```
from django.shortcuts import render
from django.http import HttpResponse

#1,helloworld
def hello_world(request):
    return HttpResponse("helloworld")
```

    - 2,创建urls.py文件,编写子应用路由

```
from django.conf.urls import url
from . import views

urlpatterns = (
    url(r'^hello/$',views.hello_world)
)
```

- 3,将子应用的路由,注册到根应用的urls中

```
from django.conf.urls import url,include
from django.contrib import admin

urlpatterns = (
    url(r'^admin/', admin.site.urls),
    url(r'^', include('users.urls')),
)

```

- 注意点:
  - 只要提供了子应用路径, django不在提供默认的根路径

4,项目的配置(settings.py文件)(理解)

- 目的: 知道常见的配置作用即可
- 常见的配置有:
  - BASE_DIR: 项目在操作系统中的绝对路径
  - DEBUG: 调试默认,默认是True
  - LANGUAGE_CODE: zh-hans中文
  - TIME_ZONE: Asia/Shanghai中国时间
- 注意点:
  - settings.py表示项目运行的配置项(调试模式,mysql,redis,session...等等)

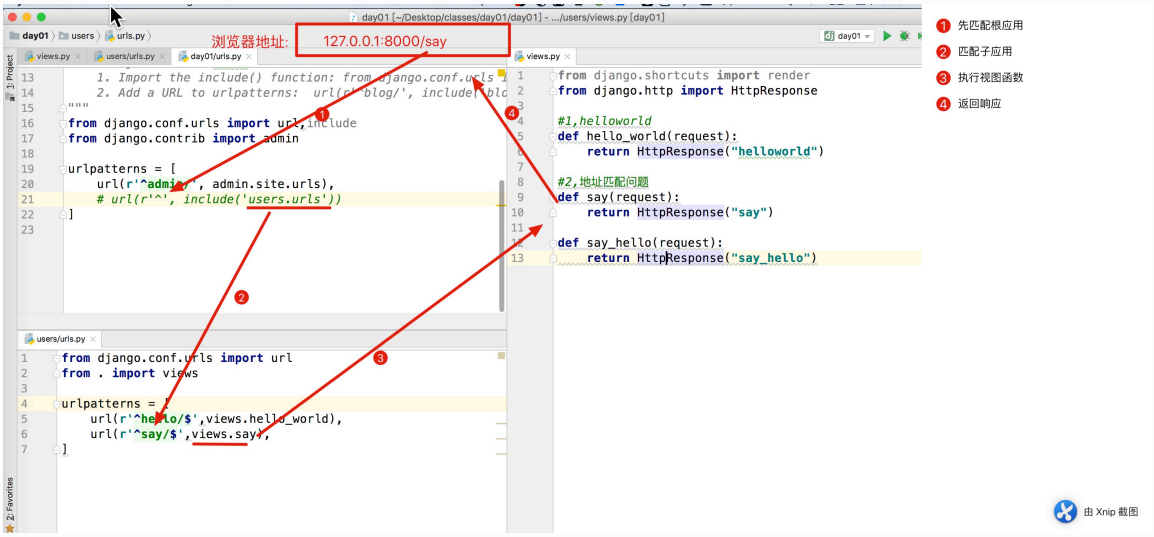
5,静态资源访问(理解)

- 目的: 在django中如何配置静态文件的访问地址和文件夹

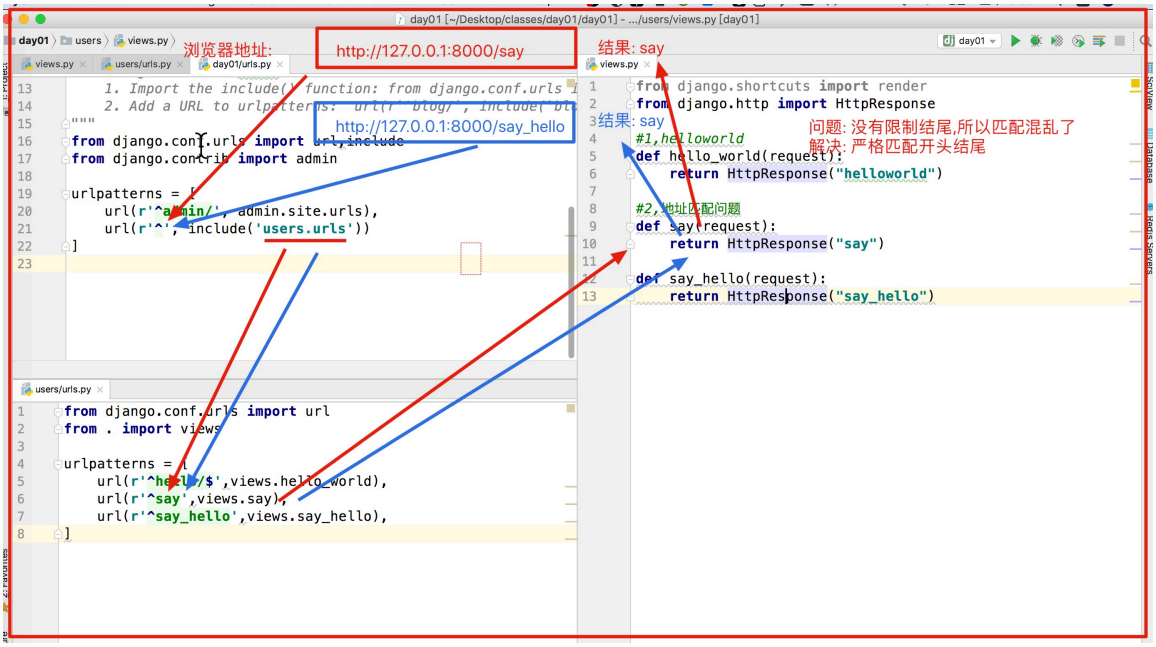
- 配置:
    - STATIC_URL: 静态资源的访问路径,默认是/static/
    - STATICFILES_DIRS: 静态资源的文件夹位置,并且是一个列表
- 注意点:
    - STATICFILES_DIRS,是一个列表,寻找文件的时候,从前向后依次寻找,找不到报404错误

6,地址匹配

- 目的: 在浏览器访问服务器的时候, 根应用的地址, 子应用的地址是如何进行匹配的
- 格式1: 基本匹配



- 格式2: 匹配混乱问题
-



- 格式3:
    - 子应用地址右面的 /, 问题, 建议大家写的时候都加上
        - 在访问的时候,浏览器自动会定向带有/的地址上面去, 这样在访问的时候不管是否有携带/, 都能访问

7, 路由参数名设置(掌握)

- 目的: 能够在地址中编写正则匹配参数
- 过程:
    - 1, 格式1,不指定正则匹配的名字: /hello/((a-z)+)/(\d+)
        - 按照顺序依次匹配,传递参数
    - 2,格式2,指定正则的名字: /hello/(?P(a-z)+)/(?P\d+)
        - 按照名字来匹配,传递参数

8, reverse

- 目的: 能够通过函数的名字name,找到对应的路径
    - 格式1:没有指定子应用的namespce, 必须指定视图函数的name
        - reserve("name")
    - 格式2:指定子应用的namespce, 必须指定视图函数的name (建议使用)
        - reserve("namespace:name")
- 注意点:
    - reverse: 称为泛解析, 通过namespace或者name得到视图函数的路径

8,查询参数

- 目的: 通过request对象,获取查询参数
- 过程:
    - request.GET : 是一个字典
    - request.GET.get(key) #获取单个key, value
    - request.GET.getlist(key) #获取单个key,对应的多个 value,得到的是list列表

9,非查询参数

- 目的: 获取request获取表单, 非表单提交的非查询参数数据
- 常见的请求方式有: POST,PUT,DELETE,PATCH
    - request.POST: 获取表单数据
    - request.body: 获取json数据
        - 需要将bytes类型转成dict格式,用到 decode(), loads()

10,其他属性

- 目的: 通过request对象获取,请求方式,请求cookie,请求用户等信息
    - request.method
    - request.users: 如果登陆过后,就是登陆的用户, 没有登录则是AnonymousUser
    - reqeuste.COOKIES

11, 响应

- 目的: 需要了解django中常见的响应对象, 能够设置响应json数据
    - 常见的响应对象
        - HttpResponse()返回文本信息的
        - JsonResponse()返回json数据的
            - safe: 指定之后可以设置列表信息为False
        - HttpResponseForbidden: 禁止访问
    - 通用的属性
        - content_type: 设置响应类型
        - status: 状态码信息

12, 重定向redirect(理解)

- 目的: 能够使用redirect重定向到远程地址,本地服务器地址, 和reverse配合使用
- 格式:
    - redirect(地址)

    - 地址: 可以是远程的, 本地的, 配合reverse使用

13, cookie(掌握)

- 目的: 能够通过request,reponse获取cookie,设置cookie内容
- 场景: 广告推送, 记录上次的浏览时间
- 操作:
    - 设置cookie
        - response.set_cookie(key,value,max_age)
        - max_age: 有效期,单位是秒
    - 获取cookie
        - request.COOKIES.get(key)

14, session(理解)